 ISO/IEC JTC 1/SC 29/WG 3 N00898

**ISO/IEC JTC 1/SC 29/WG 3**

**MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title: Draft Text of ISO/IEC DIS 23009-9 Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 9: Redundant encoding and packaging for segmented live media (REaP)**

**Status:** Approved

**Date of document:** 2023-29-04

**Source:** ISO/IEC JTC 1/SC 29/WG 3

# Expected action: ACT

**Action due date:** 2023-19-05

**No. of pages:** 43 (with cover page)

**Email of Convenor:** young.L@samsung.com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**ISO 23009-9:2023**

ISO JTC 1/SC 29/WG 3

Date: 2023-01-27

Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 9: Redundant Encoding and Packaging for segmented live media (REaP)

Draft DIS stage

**Warning for WDs and CDs**

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© ISO 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO’s member body in the country of the requester.

ISO copyright office

CP 401 • Ch. de Blandonnet 8

CH-1214 Vernier, Geneva

Phone: +41 22 749 01 11

Email: copyright@iso.org

Website: www.iso.org

Published in Switzerland

Contents

[Foreword vii](#_Toc135661360)

[Introduction viii](#_Toc135661361)

[1 Scope 1](#_Toc135661362)

[2 Normative references 1](#_Toc135661363)

[3 Terms and definitions 1](#_Toc135661364)

[3.1 Terms 1](#_Toc135661365)

[3.1.1 Media Presentation 1](#_Toc135661366)

[3.1.2 Distribution Encoder 1](#_Toc135661367)

[3.1.3 Packager 1](#_Toc135661368)

[3.1.4 Origin 1](#_Toc135661369)

[3.1.5 Unix Epoch 1](#_Toc135661370)

[3.1.6 Ceil() 1](#_Toc135661371)

[3.1.7 Storage Track File 2](#_Toc135661372)

[3.2 Abbreviated terms 2](#_Toc135661373)

[4 Document organization 2](#_Toc135661374)

[5 Reference workflow for Redundant encoding and packaging (REaP) 4](#_Toc135661375)

[5.1 Reference workflow for REAP 4](#_Toc135661376)

[5.2 REaP Objectives 5](#_Toc135661377)

[5.3 REaP Assumptions 5](#_Toc135661378)

[6 Formats for Redundant distribution Encoding 6](#_Toc135661379)

[6.1 General workflow and operation 6](#_Toc135661380)

[6.2 Constraints on the Ingest Media Presentation Description (I-MPD) 7](#_Toc135661381)

[6.3 Constraints on the redundant encoder track and segment format 8](#_Toc135661382)

[6.4 Additional Encoder Configuration aspects for delayed input and file inputs 9](#_Toc135661383)

[6.4.1 Encoder configuration using D and STS for delayed live inputs 9](#_Toc135661384)

[6.4.2 Mixed File and Live Inputs to the Encoder 9](#_Toc135661385)

[7 Formats for Redundant media presentation packaging 10](#_Toc135661386)

[7.1 General requirements on redundant packager input 10](#_Toc135661387)

[7.2 General requirements on the redundant packager output format(s) 11](#_Toc135661388)

[7.3 Specific requirements on the redundant packager D-MPD output format 11](#_Toc135661389)

[7.4 Specific requirements on the redundant packager HLS output format 12](#_Toc135661390)

[8 Asset storage and recording Format 13](#_Toc135661391)

[8.1 General 13](#_Toc135661392)

[8.2 Track format for storage of live archives 13](#_Toc135661393)

[8.2.1 Storage Track File 13](#_Toc135661394)

[8.2.2 Storage Track Identifiers 13](#_Toc135661395)

[8.3 Storage Media Presentation Description (S-MPD) 15](#_Toc135661396)

[8.3.1 Overview 15](#_Toc135661397)

[8.3.2 Constraints on the S-MPD for 24x7 live archiving and recording 16](#_Toc135661398)

[Annex A Example Media Presentation Description 17](#_Toc135661399)

[Annex B Example Segment Durations 27](#_Toc135661400)

[B.1 Recommended segment durations for encoder synchronization (informative) 27](#_Toc135661401)

[Annex C Example Applications 28](#_Toc135661402)

[C.1 Redundant and high availability distribution: 28](#_Toc135661403)

[C.2 Distributed encoding: 28](#_Toc135661404)

[C.3 A/B watermarking: 28](#_Toc135661405)

[C.4 Metadata Sources: 28](#_Toc135661406)

[C.5 Live Archives 28](#_Toc135661407)

[Annex D Example methods of carriage of Source timing (Informative) 29](#_Toc135661408)

[Annex E Example method for redundant encoder tracks generation 30](#_Toc135661409)

[Annex F Example method for generating a recording S-MPD from an I MPD 32](#_Toc135661410)

[Annex G Exchanging synchronization information between distribution encoders (informative) 34](#_Toc135661411)

[Bibliography 35](#_Toc135661412)

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](https://www.iso.org/directives-and-policies.html)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](https://www.iso.org/iso-standards-and-patents.html)).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see [www.iso.org/iso/foreword.html](https://www.iso.org/foreword-supplementary-information.html).

This document was prepared by Technical Committee ISO/TC *[or ISO/PC]* JTC1, *Information technology*,

Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO 23009 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](https://www.iso.org/members.html).

Introduction

This standard defines a reference architecture for redundant encoding and packaging of segmented live media.

In addition, it specifies format constraints to enable encoding and packaging of interchangeable media segments and media presentation descriptions.

This standard also specifies formats for 24x7 live recording and archiving of segmented live media.The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from the patent database available at [www.iso.org/patents](http://www.iso.org/patents).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO shall not be held responsible for identifying any or all such patent rights.

# Scope

This document defines formats for redundant encoding and packaging of live segmented media.

Related formats for media ingest and asset storage are also defined.

In addition, a reference workflow for using these formats for redundant encoding and packaging in practice is introduced.

# 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 23009-1, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*

ISO/IEC 23000-19, *Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media*

IETF RFC 8216: *Roger Pantos and W. May:* HTTP Live Streaming

# 3 Terms and definitions

For the purposes of this document, the terms and definitions given in [ISO/IEC 14496-12], [ISO/IEC 23009-1], [ISO/IEC 23000-19] and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

* ISO Online browsing platform: available at <https://www.iso.org/obp>
* IEC Electropedia: available at <https://www.electropedia.org/>

## Terms



### Media Presentation

CMAF presentation, DASH Media Presentation description, or HTTP multi variant live streaming playlist

### Distribution Encoder

Entity or computational unit used to encode or transcode an input to one or more ISO BMFF tracks

### Packager

Entity or computational unit used to package an input to an ISO BMFF track format and/or media streaming format such as DASH or HLS

### Origin

Entity or computational unit used to serve media content based on HTTP requests

### Unix Epoch

Seconds elapsed since 00:00:00 UTC on January 1st 1970 excluding leap seconds

### Ceil()

Nearest integer larger than or equal to the input number

### Storage Track File

File that follows constraints from ISO/IEC 23000-19 clause 7.3.3 with the exception that the earliest video media sample may not be zero and that there may be one or more SegmentIndexBoxes following first CMAF fragment.

## Abbreviated terms

The following abbreviated terms are used in this document

AF Adaptation Field

CMAF Common Media Application Format as defined in ISO/IEC 23000-19.

DASH MPEG-DASH as defined in ISO/IEC 23009-1.

DASH-IF DASH Industry Forum.

D-MPD Delivery Media Presentation Description

EBP Encoder Boundary Point

HLS HTTP Live Streaming as defined in IETF RFC 8216.

ID Identifier

IETF Internet Engineering Task Force

I-MPD Ingest Media Presentation Description

ISO BMFF ISO Base Media File Format

MPD Media Presentation Description

PID Packet Identifier

PTP Precision Time Protocol

REaP Redundant Encoding and Packaging

S-MPD Storage Media Presentation Description

STS Synchronization Time Stamp

TEMI Timeline and External Media Information

TS Transport Stream

UUID Universally Unique Identifier

VANC Vertical Ancillary Data Space

VBI Vertical Blanking Interval

PTP Precision Time Protocol

# Document organization

This specification is organized as follows: Clause 1 to 4 provide the introduction, scope, references, definitions, functions and abbreviated terms.

Clause 5 describes a reference workflow/architecture for redundant encoding and packaging of live segmented media. In addition the objectives and assumptions are detailed.

Clause 6 specifies formats for redundant encoder synchronization.

Clause 7 specifies formats for redundant packager synchronization.

Clause 8 specifies the track format and MPD constraints for asset storage

Annex A provides example media presentation descriptions for the ingest and storage applications.

Annex B provides recommendations for media segment durations.

Annex C provides example applications and use cases.

Annex D provides different ways a contribution signal may carry timing information.

Annex E provides an example method for generation of redundant encoder tracks

Annex F provides an example method for generating a recording S-MPD from an I-MPD

# Reference workflow for Redundant encoding and packaging (REaP)

## Reference workflow for REAP

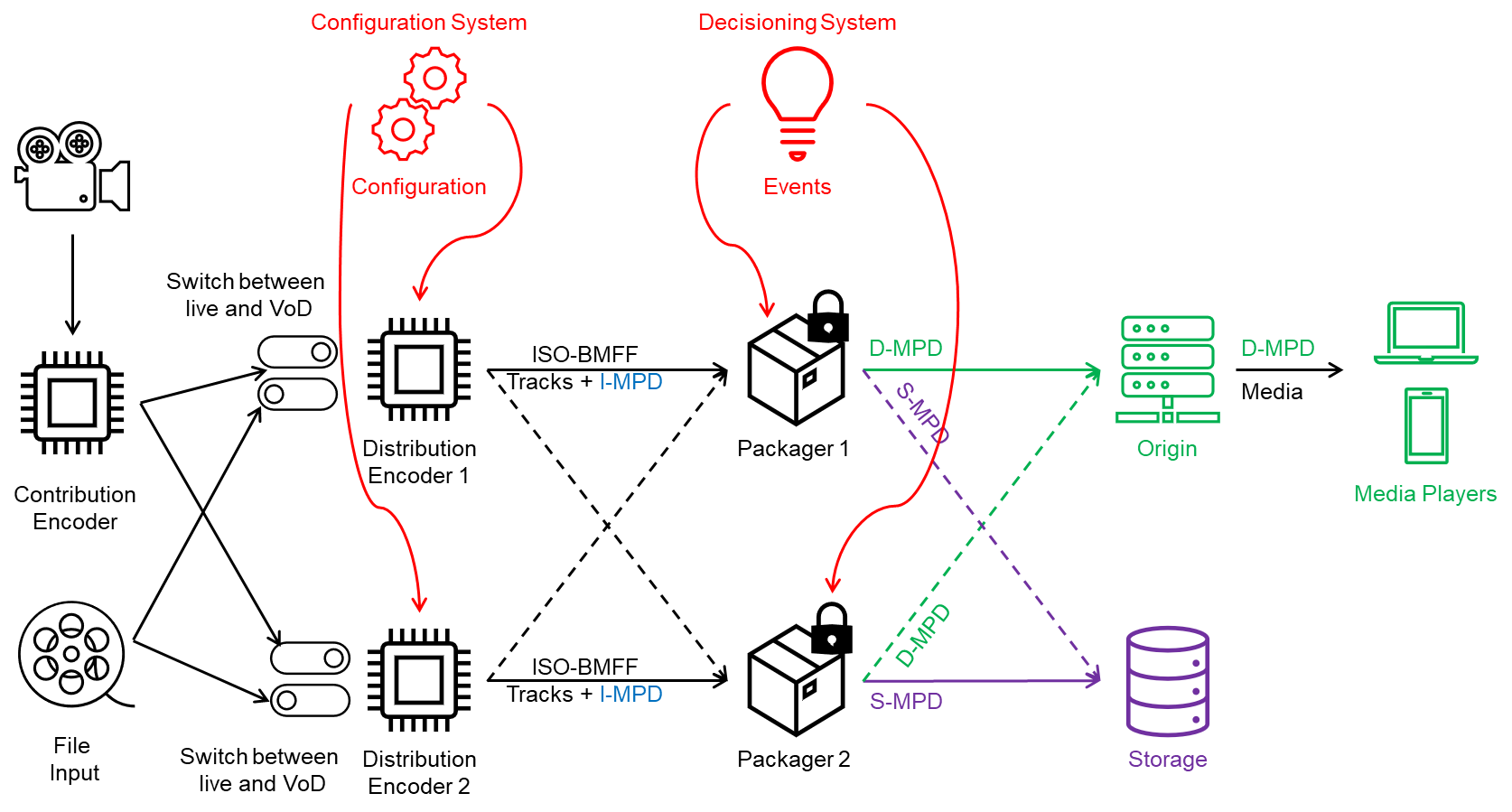


Figure 1 Reference workflow for redundant encoding and packaging of live segmented media

Figure 1 illustrates a reference workflow for redundant encoding and packaging (REaP). The workflow in Figure 1 is an example of how the formats defined in clauses 6-8 may be used. The usage of these formats is in not limited to usage as depicted in Figure 1, but Figure 1 illustrates a workflow as may be deployed in practice to enable redundant encoding and packaging.

A contribution encoder or other source transmits a signal to 2 (or more) distribution encoders that generate live segmented media. The 2 (or more) distribution encoders transmit live segmented media to 2 (or more) packagers that generate media presentations and transmit those through 1 or more origins to players or devices. In some setups the distribution encoder may combine a file input with a live input from a contribution encoder, in this case the transcoded output is a combination of the file and live input.

In addition, streams may be recorded or stored in a distributed storage.

Figure 1 details the assumptions and intended usage of the defined formats in the rest of this document as follows.

The definition of formats produced by the contribution encoder are out of scope of this document. However, it is assumed that such formats contain a common time information signal. Example formats from the contribution encoder and relevant common timing formats are given in annex D.

The redundant distribution encoders 1 and 2 generate multiple representations of segmented media. Clause 6 defines the ingest media presentation description (I-MPD) for announcing these representations. In addition, it defines the segment format constraints for the transmission of live segmented media in ISO BMFF Tracks.

The packagers 1 and 2 are responsible for generating the Delivery Media Presentation Description (D-MPD) for delivery to 1 or more clients through 1 or more origins. Constraints on the HTTP live streaming playlist, D-MPD based and live media segments are defined in clause 7. In addition, packagers 1 and 2 may apply additional transformations on the ISO BMFF tracks, such as trans-multiplexing, encryption, decryption or other operations to prepare the content for delivery.

A license server may be present, to enable a packager to request a license and decrypt incoming tracks. An encryption key server may be present, the packager can use such a server to request documents for obtaining an encryption key(s) for an incoming track or presentation in case the packager applies a (re-)encryption.

A decisioning systems may be present to insert timed events by a packager or encoder. The interface to the decisioning system is out of scope for this document, but it is expected that redundant packagers/encoders will receive functionally equivalent input from the decisioning system. The configurator is a component that may exist to configure the distribution encoders and packagers, the interfaces to the configurator are not in scope of the current document.

The functional block for storage is used to record and store media. Clause 8 defines the S-MPD (Storage MPD) to support recording and storage of redundant live segmented media.

Annex C describes some of the intended use cases of the format defined in REaP.

## REaP Objectives

REaP aims to help implementers of streaming head ends and devices to enable redundant encoding and packaging. It aims to define formats and in a reference/example workflow described in the previous clause.To achieve this the following objectives have been defined:

a) Define formats for Interchangeable Live Media Ingest and stream announcement (ingest)

b) Define aformat and segmentation strategy to generate interchangeable segments

c) Define formats for generating interchangeable media presentation descriptions or playlists

d) Define formats for efficient cloud storage access and archiving of live segmented media

e) Support approaches failover support and rejoining of distributed components in the workflow (see annex E)

f) Support workflows for live with Dynamic Ad Insertion (DAI)

g) Support workflows with DRM and content protection

h) Enable mixing file and live inputs

i) Define configuration using Synchronization Time Stamp and duration D of the distribution encoders

j) Identify and describe use cases and examples of REaP Formats

To achieve this, this document defines constraint to formats defined in ISO/IEC 14496-12, ISO/IEC 23009-1, ISO/IEC 23000-19 and IETF RFC 8216.

## REaP Assumptions

The implementation of the reference workflow with the formats defined in this document enable redundant encoding and packaging. However, it is based on several assumptions that should hold in order to achieve redundant encoding and packaging.

a) Common contribution signalling with timing information is present, e.g. see Annex E

b) Timing information can be mapped back to a time relative to Unix Epoch by configuration of synchronization time stamp (offset) (relative to 1-1-1970 excluding leap seconds)

c) Approximate clock synchronization within +- 100 ms is available on distributed entities in the workflow (stricter clock synchronization is not assumed or required )

d) Cross transmission is possible between distribution encoders and packagers and origins.

f) Distributed workflows such as with multiple data centers may be used to deploy REaP

# Formats for Redundant distribution Encoding

## General workflow and operation



Figure 2 An example workflow for redundant distribution encoder and packager operation.

Figure 2 illustrates an example of using redundant distribution encoders that follows the general reference workflow of Figure 1. In this Figure, the origin and packager functional entities are combined for simplicity.

The example workflow shows the output of the (distribution) encoders: interchangeable media segments S1, S2 etc. The segments are transmitted to packagers/origins and the output of these entities in Figure 2 is a media presentation such as based on DASH Media Presentation Description or an HTTP Live Streaming playlist.

To achieve redundancy, the segment duration D may be set as part of the encoder configuration. In addition, an optional synchronization time stamp (STS) (see Annex E) may be configured to compute the offset relative to epoch based on input timestamps.

Distributed Encoders have the same values set for these parameters and interchangeable segments have the same duration and the same earliest presentation time relative to Unix Epoch.

Specific protocol details of the transmission from encoder to packager/origin are out of scope of this document. The protocol is assumed to comprise the following steps:

1. The receiver packager sets up a URL endpoint that the distribution encoder can issue requests to.
2. The encoder authenticates, connects, and sends an ingest media presentation description (I-MPD) and initialization segments for each segmented live media representation. The I-MPD defines the grouping of Representations using AdaptationSets and the segment URL naming via SegmentTemplate elements.
3. The receiver reads the I-MPD. When the CMAF profile for DASH is used, or when content is formatted according to the CMAF track format, representations map to CMAF tracks, and the Adaptation Sets map to CMAF switching sets.
4. The distribution encoder sends each media segment using a separate request. The URL is derived from URL endpoint (Step 1) combined with the segment name derived from the SegmentTemplate Element.
5. Retransmission of media segments in a track/representation may happen in case of failures. In such cases it is recommended to retransmit the initialization segment.
6. The media segment's earliest presentation time is used as a “key” of segments in a track, the track name identified by Representation@id is used as a key for the track. Using these keys, interchangeable segments for different tracks from different senders can be identified by the receiver.

## Constraints on the Ingest Media Presentation Description (I-MPD)

The I-MPD is used to establish the naming convention of the segment URLs and grouping of tracks in adaptation sets.

The I-MPD conforms to the Media Presentation Description as defined in ISO/IEC 23009-1. Additional constraints on the I-MPD are defined as follows:

1. The I-MPD should follow the CMAF profile and iso live profiles, the MPD@profiles should include *urn:mpeg:dash:profile:cmaf:2019* and the ISO live profile of DASH *urn:mpeg:dash:profile:isoff-live:2011,* in this caseeach representation in the I-MPD represents a CMAF track, each AdaptationSet in the I-MPD represents a CMAF switching set.
2. The Period@availabilityStartTime should be set to “1970-01-01T00:00:00Z” and the Period @start should be set to PT0S or a semantically equivalent value. If this is not the case, the SegmentTemplate@presentationTimeOffset shall be used to achieve epoch relative timing instead.
3. The I-MPD shall contain a single Period element.
4. The Period@BaseURL element shall be absent.
5. EventStream elements shall not be used to signal time varying metadata if there is no frequent update of the I-MPD.
6. The I-MPD shall contain a SegmentTemplate element in each AdaptationSet element but none shall be present in a Representation Element.
7. The AdaptationSet@BaseURL element shall be absent.
8. In the case an AdaptationSet signals encrypted tracks, and decryption is required, the ContentProtection element shall be present with the following constraints:
   * 1. The ContentProtection element should contain a URL of a license server.
     2. The URL of the license server should start with https://, in that case TLS is used

NOTE: DRM used by packagers to decrypt a stream may be different from DRM used by players, a simple DRM for packagers may use a document with a decryption key included.

1. The SegmentTemplate@initialization in the I-MPD shall contain a single substring $RepresentationID$ and the SegmentTemplate@media shall contain a single substring $RepresentationID$ and a single substring $Number$ or $Time$ (not both). A separator character (- or \_) shall be between $Representation$ and $Number$ or $Time$ if one or more Representation@id ends with a number.
   1. An example of a conforming SegmentTemplate Element is: “<SegmentTemplate timescale="90000" initialization="sample-$RepresentationID$.dash" media="sample-$RepresentationID$-$Time$.dash"><SegmentTimeline> </SegmentTimeline></SegmentTemplate>”
2. SegmentTemplate@media shall be identical for each SegmentTemplate element
3. SegmentTemplate@initialization shall be identical for each SegmentTemplate element
4. Each SegmentTemplate element should contain a SegmentTimeline element
   1. A SegmentTimeline element may be empty.
5. Distribution Encoders may send an updated I-MPD
   1. In case of an updated I-MPD, identical naming conventions apply as in the previous I-MPD.
6. The I-MPD may signal requirements for encryption of an AdaptationSet
   1. In this case an EssentialProperty with the schemeIDUri: *urn:mpeg:dash:encryptionkey-server* shall be present in the AdaptationSet element.
   2. The EssentialProperty@value shall be a url to request a document from an encryption key server that describes the required encryption keys

An example of the I-MPD is given in Annex A.

## Constraints on the redundant encoder track and segment format

Entities conforming to this specification and CMAF as defined in ISO/IEC 23000-19 shall produce media segments in conformance with ISO/IEC 23000-19 (CMAF) clause 7 resulting in CMAF Tracks. Entities conforming to this specification and ISO/IEC 23009-1 shall produce media segments as described in ISO/IEC 23009-1 (DASH).

In the case of CMAF, conforming entities apply the track synchronization model specified in clause 6 as indicated in ISO/IEC 23000-19. Conforming encoders shall use the Unix epoch as the reference, therefore media presentation times shall be defined relative to 00:00:00 on January 1st 1970 excluding leap seconds.

Annex E provides an example method for redundant encoder track generation, that enables computing sequence numbers K and presentation times.

Encoder entities conforming to this specification that produce CMAF or DASH media segments shall generate boxes of the type MovieFragmentBox (moof) and optionally of the type SegmentTypeBox with the following constraints:

1. Media segments should be of a constant segment duration **D,** where the duration is the sum of all media sample presentation durations.
2. When a media segment is created with a different duration **A** instead, the next media segment shall be of duration **2 x D - A** to keep the numbering and the number of segments since epoch. This may be the case when inserting points for splicing. Alternatively, splicing may be achieved by inserting an IDR frame within the media segment and not creating variable duration segments.
3. The SegmentType box may contain a ‘slat’ brand in case the input of the encoder was missing frames and the encoder filled the gap with filler frames. In case the SegmentTypeBox contains a ‘slat’ brand, that shall be interpreted to mean that one or more frames in the segment were replaced with filler content to avoid discontinuities.
4. In case the segment is the last segment, the SegmentTypeBox should contain the ‘lmsg’ brand. If the SegmentTypeBox contains the `lmsg` brand it is the last segment in the track. The duration of the last segment may be shorter or longer than expected.
5. The sequence\_number in the MovieFragmentHeaderBox may be computed as K defined in Annex E.
6. The baseMediaDecodeTime in the TrackFragmentDecodeTimeBox may be computed as defined in Annex E.
7. An edit list, i.e. EditListBox shall not be used in the initialization segment.
8. The 'roll' sample group may be used to indicate requirements for audio playout-based pre- roll samples.

NOTE: Inserted/replaced periods that use media presentation times with a different origin (as opposed to epoch relative media presentation timing) may have an edit list, i.e. EditListBox inserted to meet requirements of the audio playout process.

1. Samples that overlap a leap second, may be adjusted to account for the leap second (reducing the duration of samples). Otherwise, media frames occurring during the leap second may be discarded by the encoder and not included in track as media sample.
2. In case the segment contains TTML text in the media segment, the TTML timing shall also be relative to the Unix epoch excluding leap seconds.

NOTE: This can result in a large number of hours in the encapsulated TTML document.

## Additional Encoder Configuration aspects for delayed input and file inputs

### Encoder configuration using D and STS for delayed live inputs

In some cases the input to the distribution encoder is delayed by several seconds. In some cases, a different anchor is used in the input timing. In these cases the output media times may not be relative to unix epoch and are not close to the current wall clock time.

In these cases the distribution encoder uses a synchronization time stamp (STS) and add this to may be shift the output media times to a timestamp relative to Unix epoch that is close to the current wall clock time.

The encoder should be configured to use D as the target segment duration, and a Synchronization Time Stamp (STS) (that defaults to 0 in this case the input timestamp maps directly to the output timestamp). The STS is added to the input timestamp (possibly accounting for the timescale conversion), to compute the output presentation time. It is recommended to choose a value of STS that enables output media presentation times close to the wall clock time or current time. The reason for this is that it enables the redundant packaging approach described later in this document.

For example, if the input value is observed to be delayed by 10 seconds compared to the current time, STS may be set to 10 seconds or more, ensuring that the output presentation time is close to current wall clock time. Configuring STS may lead to some level of buffering at the encoder as expected.

1. For configuration of D and STS the following applies:Synchronization Time Stamp STS should be configured at the encoder to produce output media presentation times close to the current time. The output timing is the sum of the input timing and STS accounted for the timescale conversion.
2. STS shall be identical at different distributed encoders as to enable distributed encoder synchronization.
3. Configuration of the target duration D shall be identical at different distributed encoders.

### Mixed File and Live Inputs to the Encoder

In some settings the encoder combines live input with file based input (see Figure 1). In this case a configuration API may be used to determine *when* to switch between live and file input. The API or configuration format of the switch is out of scope, but the switching should be based on the media timeline, which is defined as being relative to Unix epoch.

Examples of file inputs include pre-roll or interstitials inserted by the encoder. To enable seamless switching, it is recommend some STS is applied as described in the previous section. This way the file can be played out as scheduled based one the media time, and due to the delay on the live input it will be possible to switch back seamlessly to the live feed.

This way two different encoders scheduled to play the same file from 3.00 to 4.00 on date XX (for example) will generate media segments with the same segment timings (timestamps corresponding to 3.00 to 4.00 at date XX on the media timeline in ticks since epoch (excluding leap seconds)).

The following applies when combining file and live inputs to the distribution encoder.

1. Synchronization Time Stamp STS shall be configured at the different distributed encoders to produce output presentation times close to the wall clock time from the live input.

2. The file input may be transcoded to media presentation times based on the scheduled media start time and media end time of playback of the asset.

3. A third party Application Programming Interface (API) or configuration may be used to define the transition points between file and live input to the encoder based on the output media timeline. In this case the API shall trigger identical transition points at different encoders based on a target media presentation time using identical input files.

# Formats for Redundant media presentation packaging

## General requirements on redundant packager input

Additional requirements on formats used as input to redundant packagers are defined as follows:

1. As shown in Figure 1, encoders cross-transmit segments to the different available redundant packagers in the workflow. Each encoder 1……N transmits all output media segments to all packagers 1…..M. Interchangeable segments of each Representation shall have identical earliest media presentation time, for example K x D x track\_timescale.

NOTE: In some practical cases cross transmission maybe costly, optimizations in the transmission protocol and delivery are encouraged, but these are out of the scope of this document.

2. Encoders shall transmit segments within configurable bounds. Therefore, a segment with earliest presentation **T** shall be transmitted within T + segment\_duration + a bounded encoder and T + segment\_duration. Therefore, redundant packagers should receive interchangeable input media segments within bounded time differences.

NOTE: In some setups a backup redundant distribution encoder is configured to transmit at a slight delay, i.e. such as half a segment duration compared to the main distribution encoder.

3. Each encoder should write one or more ProducerReferenceTimeBoxes for each segment, when present it shall contain the time a segment entered the encoder if the flags field was set to 0 or the time it was encoded if the flags field was set to 1.

4. Splice information and metadata should be available at each of the packagers prior to the splice point time.

5. If DASHEventMessageBoxes are used they shall be repeated in segments as long as the corresponding event overlaps the segment media presentation times.

6. If a separate segmented timed metadata track is used, metadata shall be repeated in samples as long as it is applies.

## General requirements on the redundant packager output format(s)

The Packager is responsible for converting the I-MPD and live media segments to a valid delivery MPD (D-MPD) or media playlist(s) based on RFC 8216. It is assumed that the workflow from Figure 1 or Figure 2 is used where 2 (or more) distribution encoders transmit to 2 (or more) packagers.

This clause defines recommended output formats to enable synchronized and interchangeable generation of media playlists from distributed packagers.

NOTE: The D-MPD or media playlist contains relevant information for a player. This includes, for example, pre-selections, AdaptationSets and ContentProtection signaling. A packager can generate different versions of the D-MPD or playlist for different players with different requirements. The D-MPD is intended to be customized based on specific needs or requirements of the targeted players. Specific D-MPD or media playlists with the same customization generated by different packagers are interchangeable.

The following general requirements apply:

1. Media segment formatting should follow the constraints defined in clause 6.3.

NOTE: Inserted/replaced periods that use period relative media presentation times (as opposed to epoch relative media presentation timing) may have an edit list (EditListBox) inserted to meet requirements for the audio playout process, and also do not comply with segment formats defined in clause 6.3.

2. When HTTP based communication is used, upon an HTTP request or response containing a media playlist or D-MPD in the body, set the Last-Modified HTTP Header: that has a syntax of: <day-name>, <day> <month> <year> <hour>:<minute>:<second> GMT to a time corresponding to the earliest presentation plus segment duration of the newest segment.

NOTE: This avoids some of the race conditions of MPD’s generated by different redundant packagers in

The content delivery network or the player.

3. To enable debugging, the packagers should write ProducerReferenceTimeBoxes with flags set to 4 to indicate when a media segment was written to file. This may result in more than one ProducerReferenceTimeBox per segment

4. The D-MPD or media playlist based on RFC 8216 shall contain all segments that are expected to be available.

## Specific requirements on the redundant packager D-MPD output format

Apart from clause 7.2 the following additional requirements for generating a D-MPD based on ISO/IEC 23009-1 apply:

1. The D-MPD shall conform to ISO/IEC 23009-1. The D-MPD should be generated according to the mapping described in the CMAF profile for DASH 23009-1 clause 8.12. Further constraints on generation of the D-MPD may be implementation dependent and application specific.

2. The D-MPD shall set the MPD@publishTime to the time corresponding to the earliest presentation time plus segment duration of the newest segment.

3. The durations of representations across the D-MPD should be identical within a 100 milli seconds tolerance. If not identical, at least the earliest presentation time of the newest segment in different media representations shall be identical within a 100 millisecond tolerance accounting for timescale differences.

NOTE: Redundant packagers using a shared storage is one way of enabling such consistent media presentation generation between redundant packagers.

## Specific requirements on the redundant packager HLS output format

The following requirements apply for the generation of media playlists based on RFC 8216.

1. The segment URL’s indicated under an #EXTINF tag shall follow a naming structure that can be expressed using a SegmentTemplate@media string using $Number$ or $Time$.

2. Playlist shall include one or more #EXT-X-PROGRAM-DATE-TIME tags to link the wall clock time to the media segment time.

3. Media presentation timestamps of the live media segments shall be relative to Unix epoch and segment duration shall be near constant, or durations shall be compensated between subsequent segments.

4. In case the segment format is based on transport stream i.e. MPEG-2 TS, the presentation time stamps shall correspond to the media presentation time stamps from the media segments received from the encoder, but wrapped in 33 bits and using a 90 Khz scale.

5. In case MPEG-2 Transport stream is used, the #X-TIMESTAMP-MAP=MPEGTS:<MPEG-2 time>, LOCAL=YYYY-MM-DDTHH:MM:SS.mmmZ shall be used to map the MPEG-2 transport stream presentation timestamps to the corresponding local time.

6. In case of multiple media playlists, media segments should be aligned between media playlists within at least 100 milli seconds. Therefore, the earliest presentation time plus duration of newest segment in different playlist should not differ by more than 100 milli-seconds between media playlists. If this is not the case, at least the presentation time of the newest segments in different media playlists shall not differ by more than 100 milliseconds.

# Asset storage and recording Format

## General

Asset storage and recording uses the ISO Base Media File format ISO/IEC 14496-12 and a storage Media Presentation Description (S-MPD) using Media presentation description defined in ISO/IEC 23009-1.

Further restrictions to the track storage format are defined in clause 8.2. Further constraints to the Storage Media Presentation Description (S-MPD) constraints are defined in clause 8.3.



Figure 3 An example workflow for asset recording and archiving content in a cloud storage. The cloud storage may be accessed from the encoder, origin/packager or a client retrieving the media stream.

Figure 3 illustrates an example of using the specified method including cloud storage. Media presentations can be written to the cloud storge from the encoder, from the origin/packager or from a client receiving the media content. The format for asset storage and recording enables efficient and interoperable writing of content to a cloud storage, and efficient retrieval later on.

## Track format for storage of live archives

### Storage Track File

A storage track file is defined as follows.

A stored ISOBMFF track that follows the constraints of aCMAF Track file (CMAF clause 7.3.3), without the restriction that the track starts at presentation time zero. In addition, it may also contain interleaved SegmentIndexBoxes after the first MovieFragmentBox that may occur at the start of a segment to enable random access on portions of the track or segment.

The CMAF Track format structure is used to store media content as defined in clause 7 of ISO/IEC 23000-19 [2] with the exceptions as described in this clause.

### Storage Track Identifiers

This clause defines urn’s and identifiers to code properties in each track to allow recreating a media presentation or playlist if needed. Both identifiers corresponding to DASH and CMAF constructs are introduced and linked.

Figure 4 links the DASH identifiers and the CMAF identifier constructs. Table 2 defines the schemeURI for each identifier when used according to this proposal when signalling in a `kind` box. The schemeURI and value attribute in kind shall be set to the identifier scheme and UUID/ID respectively.

Track identifiers may be present in tracks. If present, they shall be signalled using a schemeURI in the kind box in UserDataBox (udta) as and the value as a UTF-8 character encoded and null terminated UUID or other type of id value as string. The defined schemeURI from Table 2 may be included in a storage track file.



Figure 4 Relationship between the manifest and its segments.



Figure 5 Orphaned content stream or asset with information to do reverse lookup.

1. Example CMAF identifiers and mapping to DASH identifiers (informative?).

|  |  |
| --- | --- |
| **DASH construct** | **CMAF construct** |
| MPD ID | CMAF Presentation ID |
| Period ID | CMAF Presentation ID |
| AdaptationSet  Group ID | Aligned Switching Set ID |
| AdaptationSet ID | SwitchingSet ID |
| Representation ID | CMAF track id (not to confuse with track\_id) |

1. Link between constructs in DASH and HLS and defined urn’s (informative)

|  |  |
| --- | --- |
| Content ID (optional) | urn:mpeg:asset-storage-format:cmaf-content-id |
| CMAF Presentation ID (MPD) | urn:mpeg:asset-storage-format:cmaf-mpd-id |
| CMAF PresentationID (Period) | urn:mpeg:asset-storage-format:cmaf-period-id |
| Aligned Switching Set ID | urn:mpeg:asset-storage-format:cmaf-aligned-switching-id |
| SwitchingSet ID | urn:mpeg:asset-storage-format:cmaf-switching-id |
| CMAF track id (not to confuse with track\_id) | urn:mpeg:asset-storage-format:cmaf-track-id |

## Storage Media Presentation Description (S-MPD)

### Overview

A manifest presentation description (MPD) can be used to reference stored track files.

NOTE: In addition to conforming to ISO/IEC 23009-1, the S-MPD contains information for generating or deriving D-MPDs or playlists for streaming clients. This includes, for example, all Pre-selections and AdaptationSets and bit-rate profiling/switching and choosing DRM systems at the server side as well as the client side. It is possible to generate different customized media presentation descriptions from the information in the S-MPD.

Table 3 summarizes the profiles and constraints for the different storage, recording and archiving options.

1. Options and profiles for Storage MPD (S-MPD)

|  |  |  |
| --- | --- | --- |
|  | @profiles | constraints |
| 24x7 live | urn:mpeg:dash:profile:isoff-live:2011 | As in clause 8.3.2 |
| simple live | urn:mpeg:dash:profile:isoff-live:2011 | may include SegmentTimeline, may use CMAF profile for DASH |
| VoD | urn:mpeg:dash:profile:isoff-on-demand:2011 | may use CMAF profile for DASH |

### Constraints on the S-MPD for 24x7 live archiving and recording

An S-MPD should be used to reference tracks used in an archived or stored/recorded media presentation.

Fixed duration track files with duration M are stored and signalled as long duration media segments in the S-MPD. These media segments contain all data from the archive track file except the initialization segment or CMAF Header. They may also include a SegmentIndexBox.

The receiver/archiver is configured to generate and store track files using a fixed duration M. The S-MPD uses the iso live profile, and SegmentTemplate elements with @duration set to the fixed duration M accounted for the timescale of the tracks or using a SegmentTemplate with a SegmentTimeline element. Each track file is stored as a media segment in the Media Presentation Description.

Table 4 specifies the constraints on the S-MPD for recording.

Annex E provides an example mapping from the I-MPD to the S-MPD.

1. Constraint on the S-MPD for recording 24x7 live content

|  |
| --- |
| 1. The MPD follows the iso live profile (urn:mpeg:dash:profile:isoff-live:2011) 2. MPD@type shall be ‘dynamic’ 3. @availabilityStartTime is set to the earliest presentation time of the first archive track segment (or if it is not known to the time the MPD was written) 4. Period@start is set to “PT0S” or a semantically equivalent value 5. MPD@timeShiftBufferDepth and Period@timeShiftBufferDepth shall not be present 6. Each AdaptationSet shall contain a SegmentTemplate element constrained as follows:    1. Each SegmentTemplate@presentationTimeOffset is present to make the media presentation timing relative to the epoch anchor, i.e. 1970-01-01T00:00:00Z, the time is MPD@availabilityStartTime - 1970-01-01T00:00:00Z adjusted for the SegmentTemplate@timescale    2. The SegmentTemplate@duration is set to the fixed duration M adjusted for SegmentTemplate@timescale    3. a SegmentTimeline may be used to indicate the start time and duration of track segments.    4. SegmentTemplate@media is the same for each SegmentTemplate and contains the substring “$RepresentationID$” and “$Number$”    5. SegmentTemplate@initialization is the same for each SegmentTemplate and contains the substring “$RepresentationID$” |

# Example Media Presentation Description

(Informative)

* 1. **Example I-MPD (informative)**

An example I-MPD is shown below.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <MPD  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="urn:mpeg:dash:schema:mpd:2011"  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH\_schema\_files/DASH-MPD.xsd"  type="dynamic"  availabilityStartTime="1970-01-01T00:00:00Z"  publishTime="2022-09-30T12:49:03.521051Z"  minimumUpdatePeriod="PT2S"  maxSegmentDuration="PT2S"  minBufferTime="PT10S"  profiles="urn:mpeg:dash:profile:isoff-live:2011,urn:com:dashif:dash264">  <Period  id="1"  start="PT0S">  <AdaptationSet  id="1"  group="1"  contentType="audio"  lang="en"  minBandwidth="64000"  maxBandwidth="128000"  segmentAlignment="true"  audioSamplingRate="48000"  mimeType="audio/mp4"  codecs="mp4a.40.2"  startWithSAP="1">  <AudioChannelConfiguration  schemeIdUri="urn:mpeg:dash:23003:3:audio\_channel\_configuration:2011"  value="1" />  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />  <SegmentTemplate  timescale="48000"  initialization="live-$RepresentationID$.dash"  media="live-$RepresentationID$-$Time$.dash">  <SegmentTimeline>  </SegmentTimeline>  </SegmentTemplate>  <Representation  id="audio\_eng=64000"  bandwidth="64000">  </Representation>  <Representation  id="audio\_eng=128000"  bandwidth="128000">  </Representation>  </AdaptationSet>  <AdaptationSet  id="2"  group="2"  contentType="video"  par="16:9"  minBandwidth="500000"  maxBandwidth="1000000"  segmentAlignment="true"  width="1280"  height="720"  sar="1:1"  frameRate="25"  mimeType="video/mp4"  codecs="avc1.42C01F"  startWithSAP="1">  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />  <SegmentTemplate  timescale="600"  initialization="live-$RepresentationID$.dash"  media="live-$RepresentationID$-$Time$.dash">  <SegmentTimeline>  </SegmentTimeline>  </SegmentTemplate>  <Representation  id="video=500000"  bandwidth="500000"  scanType="progressive">  </Representation>  <Representation  id="video=1000000"  bandwidth="1000000"  scanType="progressive">  </Representation>  </AdaptationSet>  </Period>  <UTCTiming  schemeIdUri="urn:mpeg:dash:utc:http-iso:2014"  value="https://time.akamai.com/?iso" />  </MPD> </Period>  <UTCTiming  schemeIdUri="urn:mpeg:dash:utc:http-iso:2014"  value="https://time.akamai.com/?iso" />  </MPD> |

* 1. **Example D-MPD (informative)**

An example D-MPD is shown below.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <MPD  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="urn:mpeg:dash:schema:mpd:2011"  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH\_schema\_files/DASH-MPD.xsd"  type="dynamic"  availabilityStartTime="1970-01-01T00:00:00Z"  publishTime="2022-09-30T13:49:03.521051Z"  minimumUpdatePeriod="PT2S"  timeShiftBufferDepth="PT5M"  maxSegmentDuration="PT2S"  minBufferTime="PT10S"  profiles="urn:mpeg:dash:profile:isoff-live:2011,urn:com:dashif:dash264">  <Period  id="1"  start="PT0S">  <BaseURL>dash/</BaseURL>  <AdaptationSet  id="1"  group="1"  contentType="audio"  lang="en"  minBandwidth="64000"  maxBandwidth="128000"  segmentAlignment="true"  audioSamplingRate="48000"  mimeType="audio/mp4"  codecs="mp4a.40.2"  startWithSAP="1">  <AudioChannelConfiguration  schemeIdUri="urn:mpeg:dash:23003:3:audio\_channel\_configuration:2011"  value="1" />  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />  <SegmentTemplate  timescale="48000"  initialization="live-$RepresentationID$.dash"  media="live-$RepresentationID$-$Time$.dash">  <SegmentTimeline>  <S t="79898181120000" d="92160" r="156" />  </SegmentTimeline>  </SegmentTemplate>  <Representation  id="audio\_eng=64000"  bandwidth="64000">  </Representation>  <Representation  id="audio\_eng=128000"  bandwidth="128000">  </Representation>  </AdaptationSet>  <AdaptationSet  id="2"  group="2"  contentType="video"  par="16:9"  minBandwidth="500000"  maxBandwidth="1000000"  segmentAlignment="true"  width="1280"  height="720"  sar="1:1"  frameRate="25"  mimeType="video/mp4"  codecs="avc1.42C01F"  startWithSAP="1">  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />  <SegmentTemplate  timescale="600"  initialization="live-$RepresentationID$.dash"  media="live-$RepresentationID$-$Time$.dash">  <SegmentTimeline>  <S t="998727264000" d="1152" r="156" />  </SegmentTimeline>  </SegmentTemplate>  <Representation  id="video=500000"  bandwidth="500000"  scanType="progressive">  </Representation>  <Representation  id="video=1000000"  bandwidth="1000000"  scanType="progressive">  </Representation>  </AdaptationSet>  </Period>  <UTCTiming  schemeIdUri="urn:mpeg:dash:utc:http-iso:2014"  value="https://time.akamai.com/?iso" />  </MPD> |

* 1. **Example S-MPD (informative)**

An example S-MPD is shown below.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  3 <MPD  4 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  5 xmlns="urn:mpeg:dash:schema:mpd:2011"  6 xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH\_schema\_files/DASH-MPD.xsd"  7 type="dynamic"  8 availabilityStartTime="1970-01-01T00:00:00Z"  9 publishTime="2023-03-22T15:26:11.892871Z"  10 minimumUpdatePeriod="PT2S"  11 timeShiftBufferDepth="PT30S"  12 profiles="urn:mpeg:dash:profile:isoff-live:2011">  13 <Period  14 start="PT0S">  15 <AdaptationSet  16 id="audio"  17 group="1"  18 contentType="audio"  19 segmentAlignment="true">  20 <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />  21 <SegmentTemplate  22 timescale="10000000"  23 duration="150000000"  24 startNumber="109961525"  25 endNumber="109961528"  26 media="$RepresentationID$-$Number$.cmfa">  27 <!-- 2022-04-08T13:01:02.349333Z / 1649422862 - 2022-04-08T13:02:00.866666Z -->  28 <SegmentTimeline>  29 <S t="16494228623493333" d="144213334" />  30 <S d="133546666" />  31 <S d="150613334" />  32 <S d="156800000" />  33 </SegmentTimeline>  34 </SegmentTemplate>  35 <Representation  36 id="oceans-64k"  37 bandwidth="65000">  38 </Representation>  39 </AdaptationSet>  40 <AdaptationSet  41 id="textstream\_eng"  42 group="3"  43 contentType="text"  44 lang="en"  45 minBandwidth="1000"  46 maxBandwidth="1000">  47 <Role schemeIdUri="urn:mpeg:dash:role:2011" value="subtitle" />  48 <SegmentTemplate  49 timescale="10000000"  50 duration="150000000"  51 startNumber="109961525"  52 endNumber="109961528"  53 media="$RepresentationID$-$Number$.cmft">  54 <!-- 2022-04-08T13:01:01Z / 1649422861 - 2022-04-08T13:02:01Z -->  55 <SegmentTimeline>  56 <S t="16494228610000000" d="140000000" />  57 <S d="160000000" />  58 <S d="140000000" />  59 <S d="160000000" />  60 </SegmentTimeline>  61 </SegmentTemplate>  62 <Representation  63 id="ttml"  64 bandwidth="1000">  65 </Representation>  66 <Representation  67 id="wvtt"  68 bandwidth="1000">  69 </Representation>  70 </AdaptationSet>  71 <AdaptationSet  72 id="video"  73 group="2"  74 contentType="video"  75 par="15200:6283"  76 minBandwidth="236000"  77 maxBandwidth="1340000"  78 segmentAlignment="true">  79 <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />  80 <SegmentTemplate  81 timescale="10000000"  82 duration="150000000"  83 startNumber="109961525"  84 endNumber="109961528"  85 media="$RepresentationID$-$Number$.cmfv">  86 <!-- 2022-04-08T13:01:02.376666Z / 1649422862 - 2022-04-08T13:02:00.893458Z -->  87 <SegmentTimeline>  88 <S t="16494228623766667" d="144310833" />  89 <S d="133466667" />  90 <S d="150567083" />  91 <S d="156823333" />  92 </SegmentTimeline>  93 </SegmentTemplate>  94 <Representation  95 id="oceans-250k"  96 bandwidth="236000">  97 </Representation>  98 <Representation  99 id="oceans-380k"  100 bandwidth="370000">  101 </Representation>  102 <Representation  103 id="oceans-879k"  104 bandwidth="878000">  105 </Representation>  106 <Representation  107 id="oceans-1336k"  108 bandwidth="1340000">  109 </Representation>  110 </AdaptationSet>  111 </Period>  112 </MPD> |

* 1. **Example Media Playlist**

An example media playlist according to RFC 8216 is given below

|  |
| --- |
| #EXTM3U  #EXT-X-VERSION:4  #EXT-X-MEDIA-SEQUENCE:867397718  #EXT-X-I-FRAMES-ONLY  #EXT-X-TARGETDURATION:3  #EXT-X-TIMESTAMP-MAP:MPEGTS=557701792,LOCAL=2022-10-10T12:06:56.640000Z  #EXT-X-PROGRAM-DATE-TIME:2022-10-10T12:06:56.640000Z  #EXTINF:1.92, no desc  scte35-video=500000-867397718.ts  #EXTINF:1.92, no desc  scte35-video=500000-867397719.ts  #EXTINF:1.92, no desc  scte35-video=500000-867397720.ts  #EXTINF:1.92, no desc  scte35-video=500000-867397721.ts |

# Example Segment Durations

(Informative)

## Recommended segment durations for encoder synchronization (informative)

In some cases, it is difficult to achieve segment boundary alignment. This is because audio samples in ISOBMFF contain multiple media samples (e.g. 1024 samples in an MP4 audio ISOBMFF sample) and therefore the granularity of audio and video segments may not be the same. In practice choosing values based on common frame-rates and sample rates can mitigate this problem. The table below illustrated some common options to avoid misalignment, yet it may depend on the codec impelemenation. Table 4 illustrates segment duration configurations when using samples with 1024 media samples such as used in MPEG-4 audio (aac).

This annex does not address the non-integer framerate case, but practice has shown that for M=L x D, fixed duration M can be achieved enabling a pattern in the media segment durations. This could be exploited to achieve the same mechanism to achieve consistency but based on super- segment duration M.

Newer audio codecs have the feature immediate playout frame, in this case the segment duration can be aligned with the video more easily without accounting the number of audio frames in an ISO BMFF sample.

**Table 4 Example rates, timescale and segment duration at 25 fps and 30 fps**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Media type | Frame rate / sample rate | Timescale | Frames per segment | Segment duration D |
| Video | 25 fps | 25 | L\*24 | L\*0,96 |
| Audio | 48 Khz | 48000 | L\*45 | L\*0,96 |
| Timed text | - | 1000 | 1 or more | L\*0,96 |
| Metadata | - | 1000 | 1 or more | L\*0,96 |
| Video | 30 fps | 25 | L\*48 | L\*1,6 |
| Audio | 48 Khz | 48000 | L\*75 | L\*1,6 |
| Timed text | - | 1000 | 1 or more | L\*1,6 |
| Metadata | - | 1000 | 1 or more | L\*1,6 |

# Example Applications

(Informative)

## Redundant and high availability distribution:

In this application encoder synchronization method is used to achieve setups with multiple encoders and receivers with the goalto be fault tolerant. If one encoder or receiver fails seamless switchover is possible. In this use case each distributed encoder produces the full bit-rate ladder, and similarly the redundant packagers generate the complete media presentations. In case one distribution encoder or packager fails redundant streams are available due to the cross transmission.

## Distributed encoding:

In this application different encoders generate different bit-rates of different codecs of media types. In this case for example high bit-rate Ultra High Definition resolutions are encoded on a different distributed encoder. The main benefit of using the formats and architecture described in this document is that representations will be segment aligned and synchronized without explicit communication between encoders and/or packagers. In this case encoder 1 and 2 post to all packagers. Each packager generates the complete media presentation.

## A/B watermarking:

In this application, different sources are marked with an A or a B watermark resulting in watermarked segments. Segment sequences can then be combined to generate user specific watermarks to avoid content leaking. In such case the processed and outputs tracks of encoders generating A and B watermarks can be generated using the encoder synchronization formats in this document. In this case the downstream entities can also easily apply the watermark by A/B sequencing. This makes is easy to generate the A and B versions at different distribution encoders.

## Metadata and Timed Text Sources:

Some metadata tracks such as timed text, programme metadata and/or accessibility metadata e.g. Hard of Hearing subtitles could be generated at distributed sources, possibly using different computational infrastructures, such as based on automatic translation or speech interpretation. By applying the formats disclosed in this document it is easy to synchronize such tracks back for streaming in a manifest (late binding) and no explicit communication with these third party data generators is required.

## Live Archives

By using the timing and segmentation formats from this document, stored media is linked to the Unix Epoch timeline. Segments can therefore always be mapped back to original programme times and aligned to electronic programme guides without explicit conversion requirements or explicit additional scheduling information. From such live archives it is then more easy to create catchup presentations or other presentation or align it with an electronic programme guide.

# Example methods of carriage of Source timing (Informative)

(Informative)

The source signal coming into the distribution encoder can carry timing in several format depending on the signal format. In a baseband uncompressed format, timing can be carried in the digital timing frame or though production automation system messages. In a compressed format, the elementary stream can carry timing through a timing SEI message sent with every access unit. At a container level in a packetized streaming format, timing can be carried along with each packet or box or in synchronized packet or boxes. The origin of the signal may determine the type of format that timing is carried in the source signal stream and the transcoder may need to extract the timing information depending on the origin of the source signal.

Baseband video can be carried in a digital timing frame interpreted as a string of bits in a serial digital interface (SMPTE 259M). Timing can be carried as SMPTE timecode (SMPTE 12M) in the vertical blanking interval (VBI) as vertical interval timecode with each digital video time frame. Alternatively, the baseband signal can be injected with production automation messages and be feed into an encoder (SCTE 104). These messages can contain commands like insertion commands and carry along a timestamp message that is in the form of UTC or VITC time information which is needed to create splice points in the resulting compressed stream.

Post encoder, a compressed stream can be formed with timing carried in an SEI message or through frame rate calculations.

If a Video is encapsulated in a packetized stream structure of MPEG-2 TS. The PES packet can carry timing (and IDR boundary information collectively called EBP- encoder boundary point) in either the public adaptation field though the timeline descriptor (TEMI) using an AF descriptor tag of 0x04 (ISO/IEC 13818-1 Annex U -Carriage of timeline and external media information) which can carry either NTP, or PTP timing formats. A private AF descriptor tag can also be used to convey EBP information in a more concise format with NTP timing carried in this manner (Cablelabs EBP Specification). Alternatively, a media time can be carried with a PTS timestamp which give a unique timestamp in a media stream that has a duration of under 26 hours. These type of carriage structures can be carried potentially with every access unit (aka video frame).

In MPEG-2 TS, timing can also be carried in a separate PID of the packetized stream that can carry SCTE 35 messages which can carry splice information or program event mediapoints (see SCTE 35). In SCTE 35, splice commands can be used that can carry PTS time through its splice\_info, or time\_signal constructs. In the time\_signal descriptor field that can be added additionally to splice\_info commands, UTC or PTP time can be carried along with the descriptor. If a regular timing indicator is needed, this can be done either with regularly placed time\_descriptors or through the use of regularly placed splice\_null commands.

Video Elementary streams can also be carried in an MPEG-4 File format ( ISO/IEC 14496-12) where video can be carried in ISOBMFF Boxes along with other type of boxes. One type of box in the file format stream that can carry a timing reference is the prft box that can carry an NTP time format.

# Annex E Example method for redundant encoder tracks generation

(Informative)

This annex details an example method to implement redundant distributed encoder operation using formats defined in this document. The distribution encoders are expected to receive an input signal with common per frame timing information. The assumption is that the timing information can be mapped back be relative to 1-1-1970 00:00:00 UTC excluding leap seconds.

If internal timing is not relative to such a common anchor, then a Synchronization Time Stamp (STS) is used to map input timestamps to output timestamps relative to this anchor.

The STS (Synchronization Time Stamp) is the difference between the zero time of the input signal to 1-1-1970 00:00:00 UTC excluding leap seconds.

The output frame time is calculated using:

*(1)*

Frame times lie on a continuous timeline on the track. In some cases, the input time will need to be converted from its original form to a continuous form that also matches the output timescale to derive .

In case the *track\_timescale* and the input timescale are not the same, the input time should also be adjusted for this by a timescale conversion. In this case the original input time is also multiplied by the *track\_timescale* and later divided by the original (input) timescale. In this case to avoid rounding error and encoders should only choose integer multiples of the input timescale as the output .

The STS may not directly be obtained from the encoder configuration, but may, for example, be retrieved from timing markers or other metadata markers (e.g. SCTE-35) in the input signal.

Another conversion that the encoder is required to make up for is for timestamp wraparounds to map to a continuous timeline. For example, an MPEG-2 timestamp would need to be converted to a continuous timeline even when the timestamp wraps due to overflow. This conversion is conducted by the encoder.

In equation (1), *track\_timescale* is the timescale used by the media track, where timescale is as defined in ISO/IEC 14496-12 in the mdhd (MediaHeader box).

To calculate segment boundaries equation (2) is used:

*(2)*

The segment boundary of the segment is the earliest presentation time of that segment. To calculate the number of the next segment (i.e., the next ), equation (3).

*(3)*

In equation (3) *now* is the time in seconds relative to epoch calculated using a real-time system routine. Next K is the value of K to compute the next segment boundary. It is beneficial to use the current time given by now() and use an STS to target output presentation times close to the current time. This can deal with cases such as delayed input to the distribution encoder or switching between live and file inputs.

An encoder joining a synchronized encoder session, computes Next\_K and may also compute the previous K (Next\_K - 1) and start buffering and encoding frames based on received input. Once a complete media segment is encoded with the segment boundary computed as based on (2) it can be transmitted. Joining or leaving encoders use the same segment naming, thus identical

Representation@id and SegmentTemplate@media and SegmentTemplate@initialization are used in the I-MPD generated by different encoders.

The @availabilityStartTime attribute in the I-MPD is set to 1970-01-01T00:00:00Z and the Period@start should be set to PT0S. In case Period@start is not PT0s or a semantically equivalent value, the corresponding SegmentTemplate@presentationTimeOffset attributes are used to compensate the media presentation times relative to the period start time. The media presentation times are always relative to 00:00:00 1970-01-01 UTC excluding leap seconds.

Any receiver may identify identical segments based on the baseMediaDecodeTime and the naming scheme identified from the AdaptationSet@SegmentTemplate and Representation@id. Any segment from the same Representation of track is interchangeable given that it has the same baseMediaDecodeTime. This way receivers can detect redundant/duplicate segments and/or additional representations or tracks.

# Annex F Example method for generating a recording S-MPD from an I MPD

(Informative)

This annex provides an example of how a recording S-MPD can be created from an I-MPD.

A receiver may create and write a S-MPD based on an I MPD following the steps in Table 5. A receiver may write the received segments as track files references in the S-MPD using the steps detailed in Table 6.

When the live stream stops, and no more segments are expected, the MPD@type attribute may be changed from ‘dynamic’ to ‘static’ and the SegmentTemplate@endNumber, Period@duration attributes may be set in the manifest.

1. Steps for writing segments to the S-MPD for 24x7 live content

|  |
| --- |
| 1. Initialise the S-MPD based on constraints from Table 3, read the configuration values for duration M from a memory 2. Set the MPD@availabilityStartTime and SegmentTemplate@presentationTimeOffset fields to the earliest presentation time of a received archive track file (or if this is not known to the time the MPD was written) 3. Read the I-MPD and create the corresponding AdaptationSets in the S-MPD,    1. The SegmentTimeline is not used but instead SegmentTemplate with @duration is used, @duration being set to M accounted for the SegmentTemplate@timescale    2. Representation@id matches between the I-MPD and S-MPD    3. The @startNumber is set to floor(MPD@availabilityStartTime (in seconds relative to 1970-01-01T00:00:00Z)/M)    4. SegmentTemplate@media from I-MPD but replacing $Number$ instead of $Time$ 4. Only write the S-MPD to storage if it was not already present in the storage |

1. Steps for writing the archive track files for 24x7 live content or encoder synced content (assuming the S-MPD is already written)

|  |
| --- |
| 1. Compute the next archive track boundary next\_L, the L + 1th archive track file since the established anchor 1970-01-01T00:00:00Z by next\_L = ceil(now/M), where now is the seconds based timestamp relative to Epoch or the established anchor. 2. Compute the next archive track file boundary B as next\_L \* D \* output track timescale 3. Identify from the input signal the frame time information, in this case the input is a CMAF media segment, it is the CMAF media segments earliest presentation time, potentially adjust it with a synchronisation timestamp 4. Append the incoming media segment in memory to the continue creating the archive track file 5. In case the media segment earliest presentation time plus the media segment duration equals or is greater than B goto 6 otherwise to 3 6. Upload the in memory archive track file using the SegmentTemplate URL generation substituting $RepresentationID$ for the Representation@id derived from the input track and replacing $Number$ by next\_L – 1 7. GoTo 1 |

# Annex G Exchanging synchronization information between distribution encoders (informative)

At MPEG 140 it was proposed to define a format to exchange information between distribution encoders, the message should contain the following information:

* In-media timestamp for frame PTS(*T)*
* Offset from Epoch time corresponding to PTS(T), in units of microseconds
* Offset from Epoch time to the time the frame with the in-media timestamp of PTS(T) entered the encoder buffer, in units of microseconds.
* Input error notification
* Synchronization state (acquisition period, synchronized, not synchronized)
* Frame type for frame *T*
* Input URI
* Input state
* Image descriptors (e.g., colour and edge histograms)

Details of this messaging will be developed and further input is welcome.

Bibliography

[i.1] ISO/IEC 14496-12, Information technology — Coding of audio-visual objects — Part 12: ISO base media file format

[i.2] ISO/IEC 23009-1, Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats

[i.3] ISO/IEC 23000-19, Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media

[i.4] ISO/IEC 13818-1, Information technology — Generic coding of moving pictures and associated audio information — Part 1: Systems, www.mpeg.org [MPEG-2 TS]

[i.5] ANSI/SCTE 35, Digital Program Insertion Cueing Message, www.scte.org [SCTE 35]

[i.6] ANSI/SCTE 104, Automation System to Compression System Communications Applications Program Interface (API), www.scte.org. [SCTE 104]

[i.7] ANSI/SCTE 223, Adaptive Transport Stream, www.scte.org [SCTE 223]

[i.8] OC-SP-EBP\_I01-130118, OpenCable Specifications- Encoder Boundary Point Specification, www.cablelabs.com [CL-EBP]

[i.9] SMPTE ST 12, Time and Control Code, www.smpte.org [SMPTE ST 12]

[i.10] SMPTE ST 259, Television – SDTV1 Digital Signal/Data – Serial Digital Interface, [www.smpte.org](http://www.smpte.org) [SMPTE ST 259]

[i.11] IEEE 1588, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control System, www.ieee.org. [IEEE PTP]